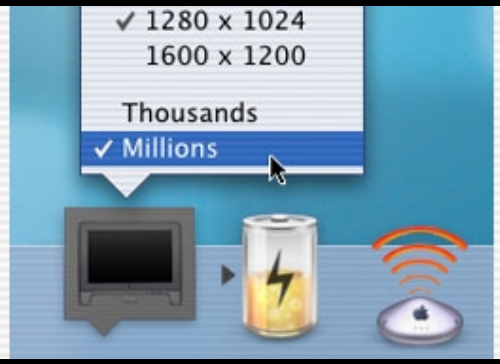
Session 701

Introducing Project Builder



Dave Payne
Manager, Development Environment Team

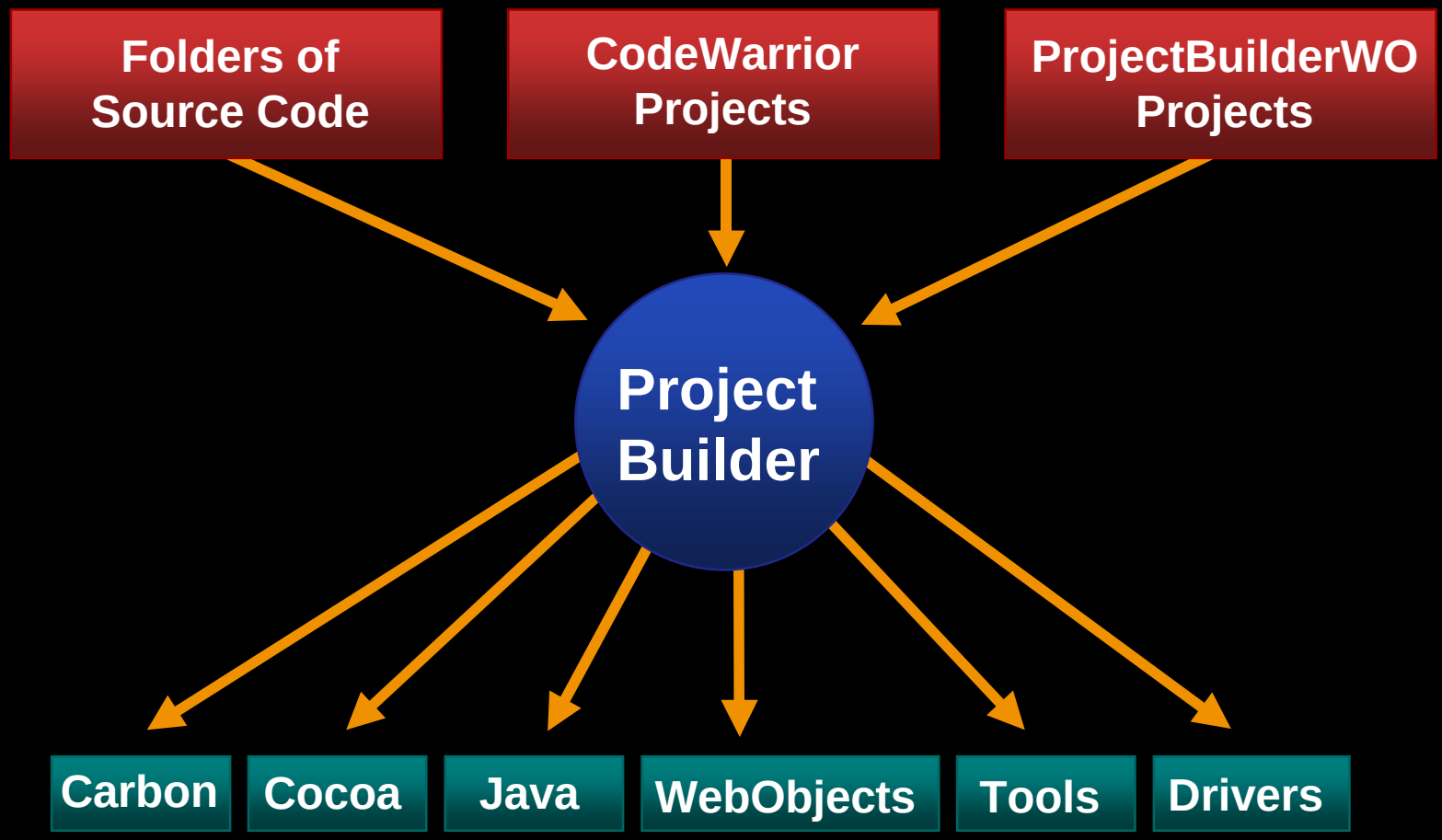
The Power of Project Builder

- What is Project Builder?
- How to use Project Builder for:
  - Creating, editing, and navigating projects
  - Source code management
  - Configuring, building, and debugging targets
- What's new on May 2001 Developer Tools CD
- The future of Project Builder

The Power of Project Builder

- What is Project Builder?
- How to use Project Builder for:
  - Creating, editing, and navigating projects
  - Source code management
  - Configuring, building, and debugging targets
- What's new on May 2001 Developer Tools CD
- The future of Project Builder

Project Builder—A Powerful IDE for All Mac OS X Development



Mac OS X Development—Project Builder Is the Hub

Interface Builder

WebObjects Builder

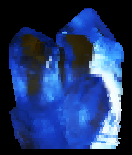
FileMerge

EOModeler

Project Builder

PropertyListEditor

PackageMaker



QuartzDebug

ObjectAlloc

Sampler

MallocDebug

Project Builder Is Industrial Strength

- Used by over 250 projects in Mac OS X at Apple
- Facilitates team development
  - App preferences for each user
  - Project settings stored in project for each user
  - Integrated with CVS Source Code Management

The Evolution of Project Builder

A testament to the power of Cocoa!

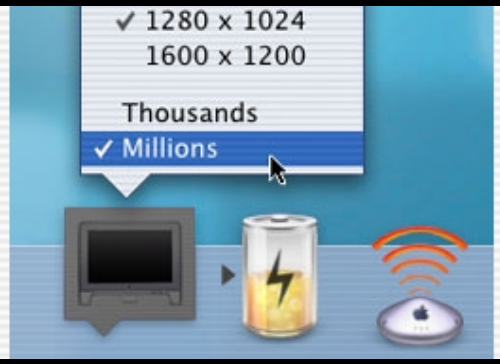
- Feb. 1999: Project kickoff
- Feb. 2000: Rollout within Apple
- March 2001: Version 1.0
- May 2001: First update—1.0.1

The Power of Project Builder

- What is Project Builder?
- How to use Project Builder for:
  - Creating, editing, and navigating projects
  - Source code management
  - Configuring, building, and debugging targets
- What's new on May 2001 Developer Tools CD
- The future of Project Builder

Session 701

A Walk Through Project Builder



Mike Ferris
Manager, Project Builder IDE Team

Overall User Interface

Project structure panes

Sketch.pbproj

Sketch

Groups & Files
- Sketch
  - View & Controller Classes
  - Model Classes
    - SKTDrawDocument.h
    - SKTDrawDocument.m
    - SKTGraphic.h
    - SKTGraphic.m
    - SKTRectangle.h
    - SKTRectangle.m
    - SKTCircle.h
    - SKTCircle.m
    - SKTLine.h
    - SKTLine.m
    - SKTTextArea.h
    - SKTTextArea.m
    - SKTImage.h
    - SKTImage.m
    - SKTRenderingView.h
    - SKTRenderingView.m
  - Other Sources
    - Sketch_main.m
  - Resources
  - Notes
  - External Frameworks and L
  - Products

Find | Build | Run | Debug

SKTRenderingView.m : -initWithFrame:graphics:

```
#import "SKTRenderingView.h"
#import "SKTGraphic.h"

@implementation SKTRenderingView

- (id)initWithFrame:(NSRect)frame graphics:(NSArray *)graphics {
    // Initialize the superclass.
    self = [super initWithFrame:frame];
    if (self) {
        _graphics = [graphics retain];
    }
    return self;
}

- (void)dealloc {
    [_graphics release];
    [super dealloc];
}

- (void)drawRect:(NSRect)rect {
    unsigned i;
```

Overall User Interface

Editor Pane and Navigation Bar

Sketch.pbproj

Sketch

Find | Build | Run | Debug

Groups & Files

- Sketch
  - View & Controller Classes
  - Model Classes
    - SKTDrawDocument.h
    - SKTDrawDocument.m
    - SKTGraphic.h
    - SKTGraphic.m
    - SKTRectangle.h
    - SKTRectangle.m
    - SKTCircle.h
    - SKTCircle.m
    - SKTLine.h
    - SKTLine.m
    - SKTTextArea.h
    - SKTTextArea.m
    - SKTImage.h
    - SKTImage.m
    - SKTRenderingView.h
    - SKTRenderingView.m
  - Other Sources
    - Sketch_main.m
  - Resources
  - Notes
  - External Frameworks and L
  - Products

Files | Bookmarks | Targets | Breakpoints

SKTRenderingView.m : -initWithFrame:graphics:

```
#import "SKTRenderingView.h"
#import "SKTGraphic.h"

@implementation SKTRenderingView

- (id)initWithFrame:(NSRect)frame graphics:(NSArray *)graphics {
    // Initialize the superclass.
    self = [super initWithFrame:frame];
    if (self) {
        _graphics = [graphics retain];
    }
    return self;
}

- (void)dealloc {
    [_graphics release];
    [super dealloc];
}

- (void)drawRect:(NSRect)rect {
    unsigned i;
```

Overall User Interface

Configurable tool bar

Sketch.pbproj

Sketch

Groups & Files

- Sketch
  - View & Controller Classes
  - Model Classes
    - SKTDrawDocument.h
    - SKTDrawDocument.m
    - SKTGraphic.h
    - SKTGraphic.m
    - SKTRectangle.h
    - SKTRectangle.m
    - SKTCircle.h
    - SKTCircle.m
    - SKTLine.h
    - SKTLine.m
    - SKTTextArea.h
    - SKTTextArea.m
    - SKTImage.h
    - SKTImage.m
    - SKTRenderingView.h
    - SKTRenderingView.m
  - Other Sources
    - Sketch_main.m
  - Resources
  - Notes
  - External Frameworks and L
  - Products

Find | Build | Run | Debug

SKTRenderingView.m : -initWithFrame:graphics:

```
#import "SKTRenderingView.h"
#import "SKTGraphic.h"

@implementation SKTRenderingView

- (id)initWithFrame:(NSRect)frame graphics:(NSArray *)graphics {
    // Initialize the superclass.
    self = [super initWithFrame:frame];
    if (self) {
        _graphics = [graphics retain];
    }
    return self;
}

- (void)dealloc {
    [_graphics release];
    [super dealloc];
}

- (void)drawRect:(NSRect)rect {
    unsigned i;
```

Overall User Interface

Tool Panes—Find, Build, Run, Debug

Sketch.pbproj

Sketch

Groups & Files
- Sketch
  - View & Controller Classes
  - Model Classes
    - SKTDrawDocument.h
    - SKTDrawDocument.m
    - SKTGraphic.h
    - SKTGraphic.m
    - SKTRectangle.h
    - SKTRectangle.m
    - SKTCircle.h
    - SKTCircle.m
    - SKTLine.h
    - SKTLine.m
    - SKTTextArea.h
    - SKTTextArea.m
    - SKTImage.h
    - SKTImage.m
    - SKTRenderingView.h
    - SKTRenderingView.m
  - Other Sources
    - Sketch_main.m
  - Resources
  - Notes
  - External Frameworks and L
  - Products

Find: initWithFrame | Default Find Options | Find

Replace: | Definitions | Replace

Whole Words | Ignore Case | Options

Found 'initWithFrame' – 3 occurrences

- SKTGraphicView.m
  - **Instance Method:** – (id)**initWithFrame**:(NSRect)frame {
- SKTRenderingView.h
  - **Instance Method:** – (id)**initWithFrame**:(NSRect)frame graphics:(NSArray
- SKTRenderingView.m
  - **Instance Method:** – (id)**initWithFrame**:(NSRect)frame graphics:(NSArray

Find | Build | Run | Debug

SKTRenderingView.m : –initWithFrame:graphics:

```
#import "SKTRenderingView.h"
#import "SKTGraphic.h"

@implementation SKTRenderingView

- (id)initWithFrame:(NSRect)frame graphics:(NSArray *)graphics {
    // Initialize the superclass.
    self = [super initWithFrame:frame];
    if (self) {
        _graphics = [graphics retain];
```

Found 'initWithFrame' – 3 occurrences

Templates for New Projects

- Applications
- Bundles
- Frameworks
- Kernel Extensions and I/O Kit Drivers
- Tools

Templates for New Projects

- Convenient starting points for new projects
- Templates contain:
  - Template source files
  - A build target
  - References to frameworks and libraries

Importing Projects

- Drag in folders of existing source code
- Import:
  - CodeWarrior projects
  - ProjectBuilderWO projects
- Importing creates group hierarchy and targets
- May need to configure targets

Groups and Files Outline

- User-definable grouping of files
  - Independent of physical location on disk
- Move files by dragging
- Group or Ungroup from Project menu
- Add Files or Frameworks by dragging in from Finder or via Project menu
  - Copies files if requested
  - Adds to selected targets

Groups and Files Outline

- Outline also gives access to:
  - Referenced frameworks
    - Header files
  - Built products
    - Package hierarchy
    - File names shown in red don’t exist on disk

Editing Files

- Syntax coloring
- Split windows
- Double-click file name for separate window
- Find –> Go to Line shows current line number

Inspecting Files—Text Settings

- Tab width
- Indent width
- End of line settings
- Language and character encodings
  - UTF8, Unicode, MacOSRoman, WinLatin1…
  - Use UTF8 to intermix languages in a C file
    - Kanji comments in source files

Inspecting Files, Groups, and Project

- Reference style
  - Group-relative
  - Project-relative
  - Absolute
- Localizations
  - Make localized or global
  - Add a localized variant
- Selecting top level of outline inspects project

Indexing a Project

- Enable indexing with Project –> Index Project
- All languages are indexed
  - Uses compiler for C, C++, Objective-C
  - Separate parser for Java
- Index is used for:
  - Symbol popup
  - Included files popup
  - Definition searches
  - More to come

Navigating Source Code

- Switching between files
  - Recently-viewed files
  - Easy access to included files
- Navigating within and between files
  - Symbol popup
  - Cmd-double-click to show definition

Bookmarks

- Direct access to anywhere in your project
- Stored on a per-user basis
- To create:
  - Select text in file
  - Choose Navigation –> Add to Bookmarks
- Rename via option-click on name

Project-Wide Find and Replace

- User-definable “find sets” control search space
  - This project only
  - All open projects
  - Frameworks
- Type of search
  - Textual
  - Regular expression
  - Definition

The Power of Project Builder

- What is Project Builder?
- How to use Project Builder for:
  - Creating, editing, and navigating projects
  - Source code management
  - Configuring, building, and debugging targets
- What's new on May 2001 Developer Tools CD
- Project future of Project Builder

Project Builder Supports CVS

Yes, but what is it?

- Concurrent Versioning System
- Open source
- Cross platform
- Comes on Mac OS X Developer Tools CD
- Apple uses it extensively
- Darwin sources are in CVS

Documentation for Using CVS in PB

- CVS Manual:
  - /Developer/Documentation/Commands/cvs/cvs_toc.html
- Project Builder's Help:
  - Projects –> Source Code Management

Preparing to Use CVS in PB

- Requires a pre-existing CVS repository
  - Local or via NFS
  - Remote via client-server with pserver or ssh
- Add project to CVS from command line
  - cvs import
- Project must be in a CVS working directory
  - Check it out from command line
  - PB looks for CVS/Root file in .pbproj directory

Committing Changed Files

- SCM –> Commit Changes…
  - On selected files or groups
  - Select top-level group to commit all changes in project
  - Makes file adds and removes permanent
- To notify CVS of new file:
  - Choose SCM –> Add to Repository
- To notify CVS of file removal:
  - Just delete it

SCM Status of Files

- Checked when group is opened
- Displayed in Groups and Files outline:

M	Modified
U	Update (has changed in repository)
?	Not in repository
A	To be Added to repository
R	To be Removed from repository
C	Conflicts merging to latest revision

Comparing and Merging Revisions

- PB launches FileMerge to visually compare and merge revisions
- SCM change logs shown in File Inspector
- SCM –> Compare with Base
  - Version that you started with
- SCM –> Compare/Merge with Latest Revision
  - Latest version on main branch in repository

Wrapped Files in CVS

- Wrapped files: directories treated as files
  - Examples: .nib, .rtfd
- To use:
  - Commit/Developer/Tools/cvswrappers into repository's top-level CVSROOT directory
  - In client/server mode, each user needs .cvswrappers file in home directory
- See Help –> Show Older Release Notes (Public Beta)

Session 702

Source Code Management Demo

Mike Ferris
Manager, Project Builder IDE Team

The Power of Project Builder

- What is Project Builder?
- How to use Project Builder for:
  - Creating, editing, and navigating projects
  - Source code management
  - Configuring, building, and debugging targets
- What's new on May 2001 Developer Tools CD
- The future of Project Builder

Targets

The “face” of the build system

- A “target” defines a product to build
- A project contains one or more targets
- Targets can depend on other targets
- “Build styles” allow variations on a target
- “Build phases” control how a target is built
- Each user has an active target and an active build style

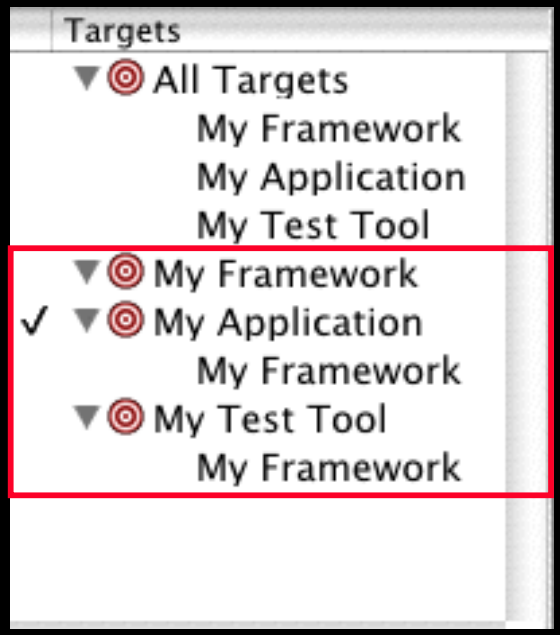
Standard Types of Targets

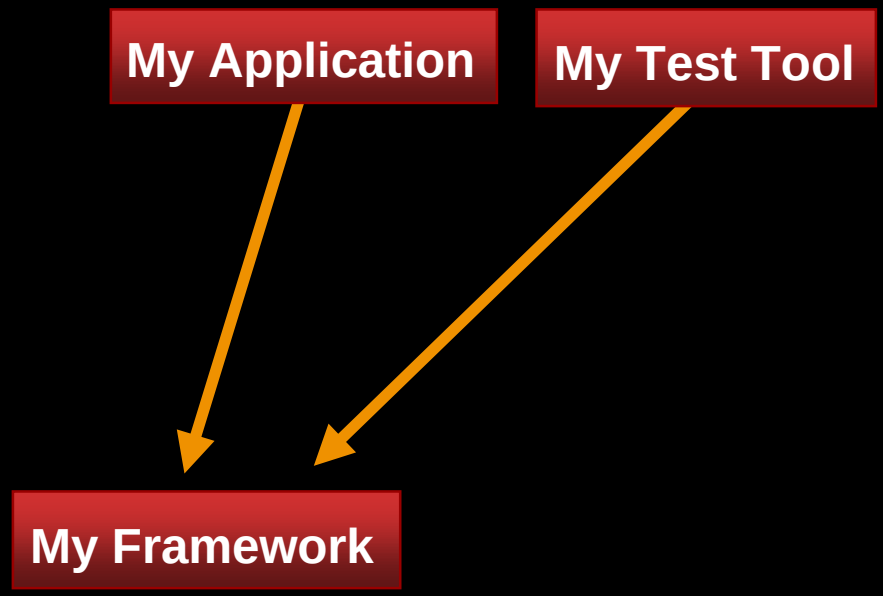
- Apps, Bundles, Frameworks, Libraries, Tools
- Java Applets and Packages
- Kernel extensions and IOKit Drivers
- Add via Project –> New Target…

Special Types of Targets

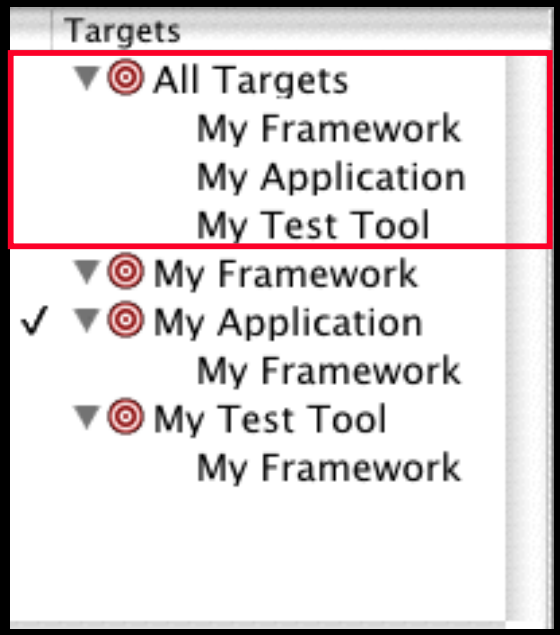
- "Aggregate" target:
  - Does not build anything itself
  - Depends on other targets
  - Useful as first target in project
- "Legacy Makefile" targets add flexibility

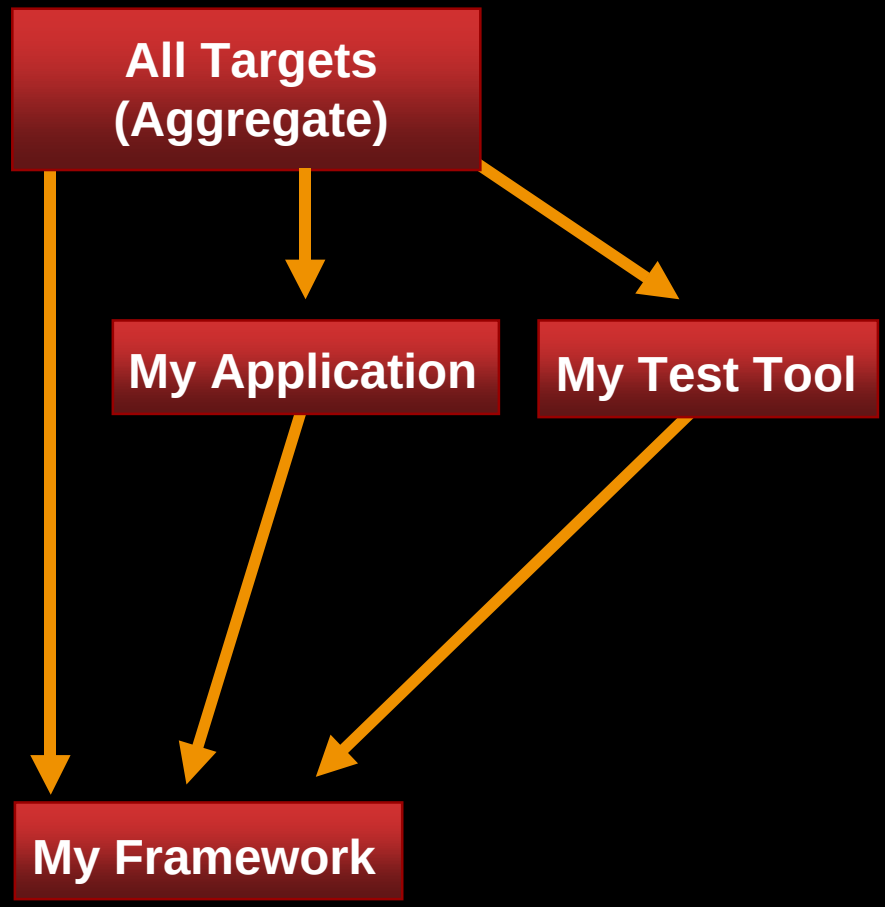
Target Dependencies





Target Dependencies



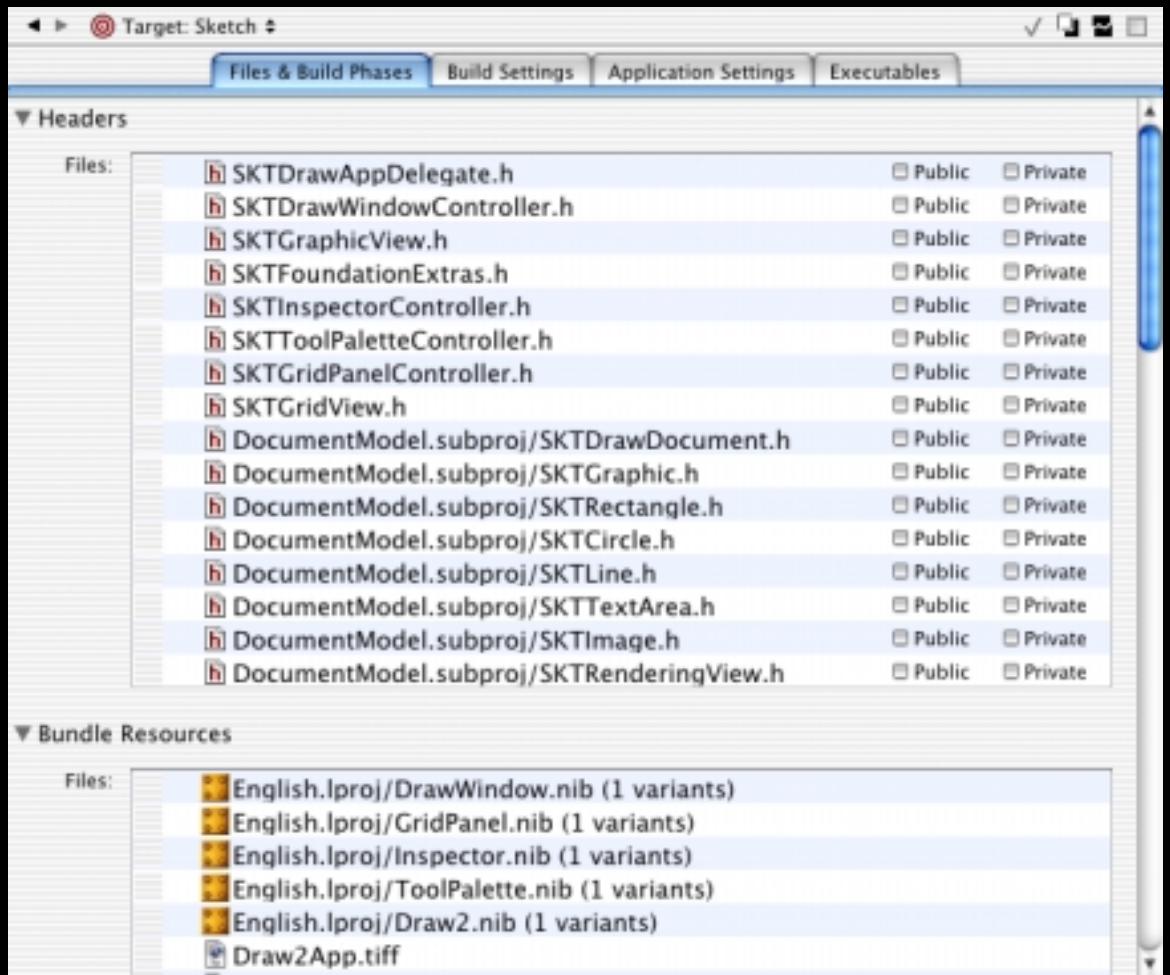


Configuring a Target

- Files & Build Phases
- Build Settings
- Bundle Settings
- Executables

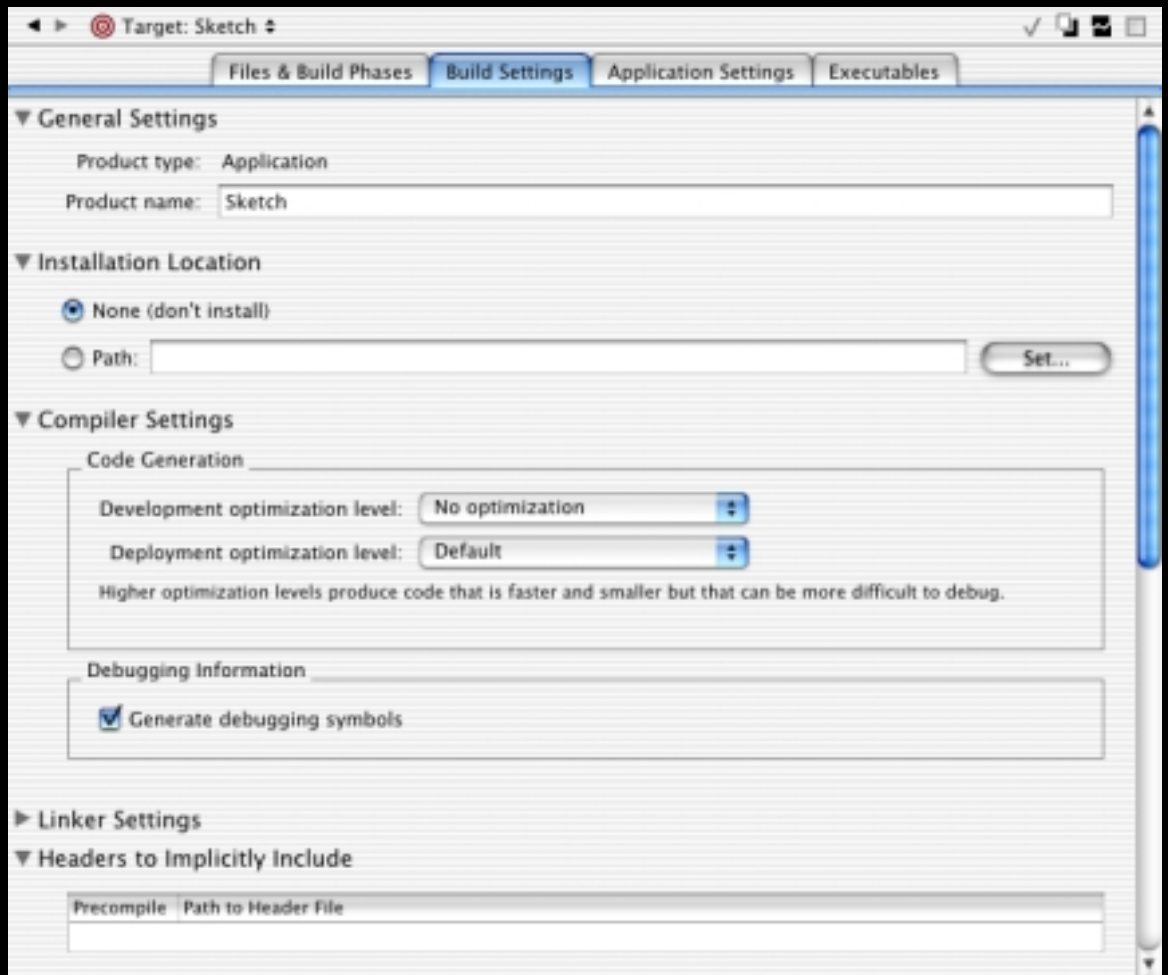
Configuring Target Build Phases

How to process files in a target



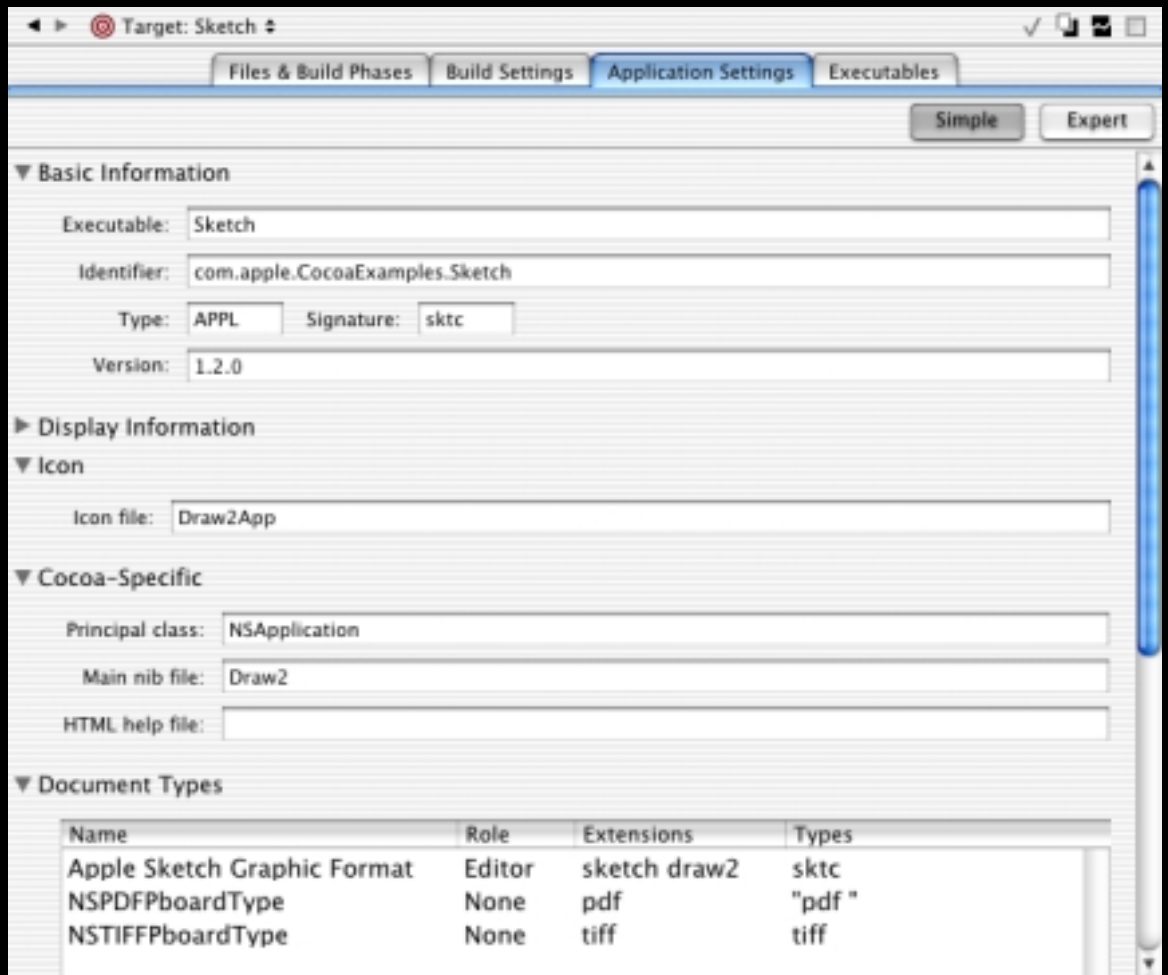
Configuring Target Build Settings

Compiler and linker options



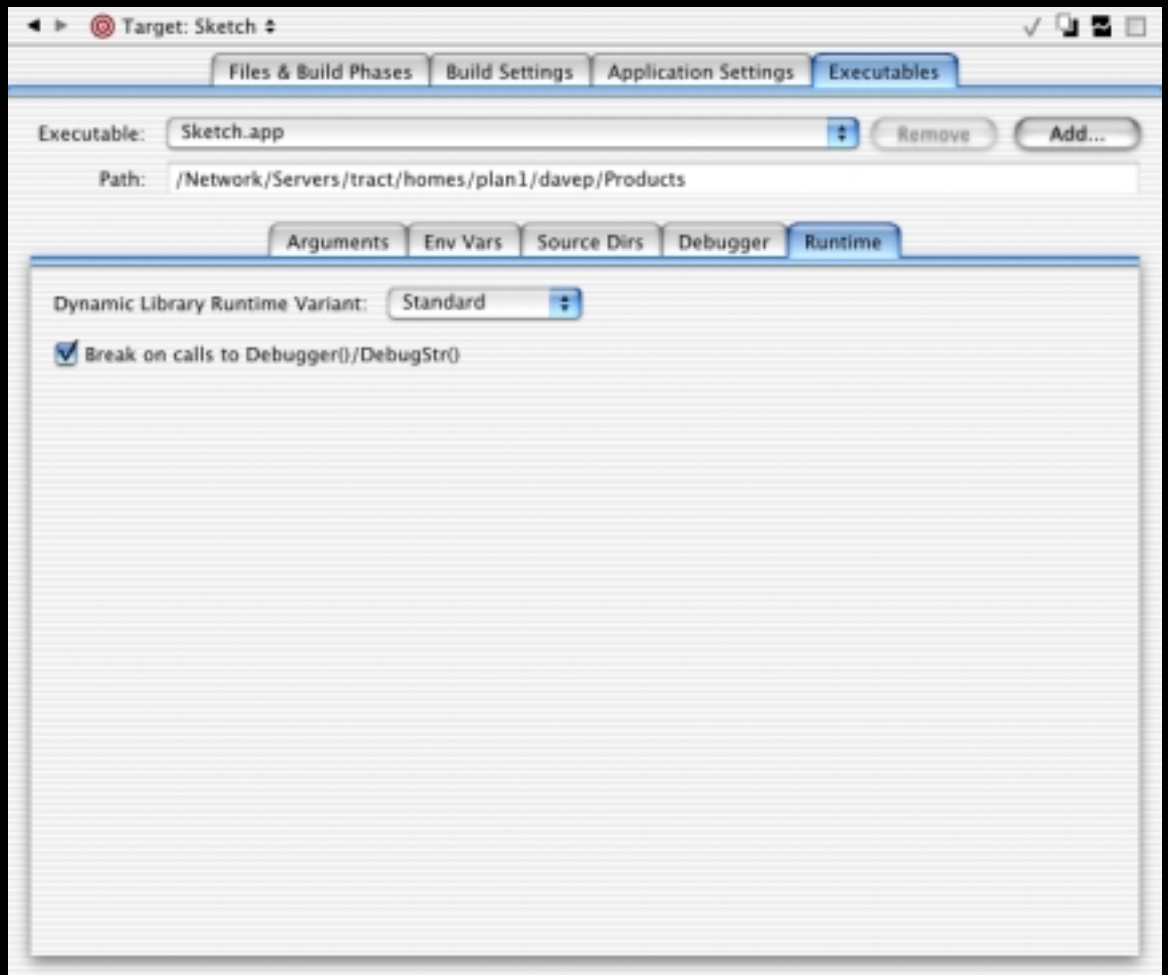
Configuring Target Bundle Settings

Info.plist for Apps, Frameworks, Bundles



Configuring Target Executables

How to run and debug



Powerful Build System

- Automatic header file dependency analysis
- No makefiles to edit
- Leverages command line utilities
  - jam build tool
  - Compilers: gcc, Rez, javac, jikes
- Complete flexibility:
  - Script build phases
  - Legacy Makefile targets

Where Do Build Results Go?

- "build" directory in each project by default
- Common build directory for all projects
  - Test apps with frameworks without installing
  - PB configures dynamic loader when running or debugging

Setting a Common Build Directory

- Per-user preference for all projects
  - In Building Preferences pane:
    - Separate location for build products
    - Separate location for intermediate build files
- Per-user preference for a set of specific projects
  - In each project:
    - Select top level of Groups & Files outline
    - Choose Project –> Show Inspector

The gcc 2.95.2 Compiler

- gcc builds all of Mac OS X
- Supports C, C++, Objective-C
  - Objective-C++ on the way!
- Velocity Engine support
- Learn more in Compiler Technologies session

C++ Language Support

- Stricter ANSI C++ compliance
- Recent performance improvements
  - Improved exception handling
  - Coalesced symbols
- C++ ABI will change with gcc 3.0!
  - Don't export C++ APIs from frameworks

Precompiled Headers

- Supplied by Apple for umbrella frameworks
- Preprocessor automatically finds precomps
- Precomps cannot contain C++ syntax, but can be included from C++ files
  - #import <Carbon/Carbon.h>
- Project Builder now supports “implicitly included headers” which can be precompiled

Building From the Command Line

- Allows automation of builds
- pbxbuild command
  - Uses same build machinery as Project Builder
  - By default, builds first target
    - An aggregate target is useful
  - Invocations look just like ‘make’

```
pbxbuild clean
pbxbuild
pbxbuild install DSTROOT=/
```

Debugging in Project Builder

- Extensible architecture:
  - gdb, Java…
- Full graphical source-level debugging
  - Threads, stack traces, variables
- Persistent, per-user breakpoints
- Controls dynamic loader for:
  - Debugging with development versions of frameworks
  - Cross-project debugging

The gdb Debugger

- Powerful open source debugger
- Handles C, C++, Objective-C
- Expression evaluation
- Invoke functions to get values of opaque types
- Commands and conditions on breakpoints
- Powerful macro capability
- Learn more in Debugger session

Session 702

Building and Debugging Demo

Mike Ferris
Manager, Project Builder IDE Team

The Power of Project Builder

- What is Project Builder?
- How to use Project Builder for:
  - Creating, editing, and navigating projects
  - Source Code Management
  - Configuring, building, and debugging targets
- What's new on May 2001 Developer Tools CD
- The future of Project Builder

May 2001 Developer Tools CD

- Full support for WebObjects 5
  - Project Builder
  - Interface Builder
- WebObjects 5 Developer included on WWDC CD
- New system documentation
- Additional examples
  - AppKit
  - Interface Builder

Project Builder Enhancements

- Localized into Japanese
- Continue build after error
- Check Syntax
- Precompiled “implicitly included header files”
- Support for HeaderDoc in build system
- Bug fixes

Enhancements for Java

- Improved Java build support
  - Merging of jar files
  - Fixes to enable building large Java projects
  - Build performance improvements
- Java wrapper build support
- “Set values” for Java debugging

The Power of Project Builder

- What is Project Builder?
- How to use Project Builder for:
  - Creating, editing, and navigating projects
  - Source Code Management
  - Configuring, building, and debugging targets
- What's new on May 2001 Developer Tools CD
- The future of Project Builder

Coming Attractions

- Faster symbol popups—we listen!
- Class browser
  - C++, Objective-C, Java
- Integrated viewing of developer documentation
- Underlying support:
  - Objective-C++
  - Two-level namespace in linker/loader

Class Browser

- How can Project Builder help me learn a class hierarchy such as Cocoa?
  - Where is this class in the class hierarchy?
  - What methods and variables are inherited?
  - Show me the declaration, implementation, or documentation of a method

Session 702

Class Browser Demo

Mike Ferris
Manager, Project Builder IDE Team

On the Drawing Board

- More flexible UI
- Plug-in editors
- Plug-in build tools
- Tighter integration with other tools
  - Direct launching of performance tools
- Compiler improvements
  - Faster C++ compilation
  - Code generation improvements

Summary

- Project Builder—a powerful IDE for all Mac OS X development
- Continual improvement via quarterly updates
- Send us your feedback
- Finish and ship your applications!!!

Information Resources

- Information pages for all Mac OS X tools
  http://developer.apple.com/tools/
- Project Builder web page
  http://developer.apple.com/tools/projectbuilder/

- Apple's mailing list master page
  http://lists.apple.com/

  Mailing lists:
  - **projectbuilder-users**
  - **cocoa-dev**
  - **carbon-dev**
  - **java-dev**

Roadmap—Tools Related Sessions

702 Interface Builder: Apple's UI design tools for Mac OS X	Hall 2 **Tues., 3:30pm (NEXT!)**
603 Introduction to WebObjects Tools	Civic **Tues., 3:30pm**
703 Project Builder In-Depth: Techniques for using Project Builder	Room A2 **Wed., 9:00am**
503 Java Development Tools: Tools for the Java developer	Room A1 **Wed., 10:30am**
114 Application Packaging & Document Binding	Hall 2 **Wed., 5:00pm**

Roadmap—Tools Related Sessions

704 Compiler Technologies: Learn about what Apple is doing in GCC	Room A2 **Thurs., 10:30am**
129 Mac OS X Application Localization	Room C **Thurs., 3:30pm**
705 Apple Performance Tools: Learn how to make your application scream!	Room A2 **Thurs., 5:00pm**
706 Debugging on Mac OS X: Debugging with Project Builder & GDB	Hall 2 **Fri., 9:00am**
707 Darwin Documentation Project & Header Doc	Room A2 **Fri., 10:30am**

Roadmap—Tools Related Sessions

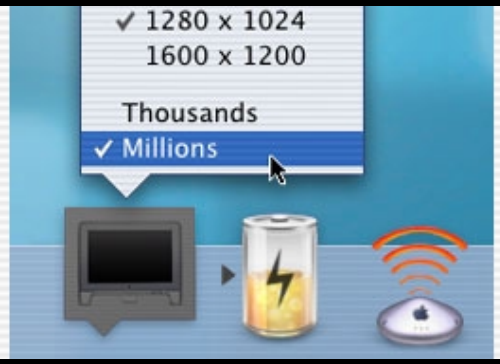
708 C++ Frameworks for Mac OS X: Take advantage for Carbon development	Room C **Fri., 2:00pm**
FF119 Apple Developer Tools: Give your feedback to the tools team!	Room J1 **Fri., 3:30pm**

Who to Contact

Godfrey DiGiorgi
Technology Manager for Development Tools
Apple Worldwide Developer Relations
ramarren@apple.com

macosx-tools-feedback@group.apple.com
Mailing address for feedback

Q&A



Godfrey DiGiorgi
Technology Manager for Development Tools
ramarren@apple.com

http://developer.apple.com/events/urls.html

Information Resources

- Information pages for all Mac OS X tools
  http://developer.apple.com/tools/
- Project Builder web page
  http://developer.apple.com/tools/projectbuilder/

- Apple's mailing list master page
  http://lists.apple.com/

  Mailing lists:
  - **projectbuilder-users**
  - **cocoa-dev**
  - **carbon-dev**
  - **java-dev**

WWDC

Worldwide Developers Conference 2001